In the United States Patent and Trademark Office

In re the application of: Kallol Pal	)	
	)	
Filed 02/14/2001	)	Group Art Unit: 2122
	)	
For: Software Testing	)	Examiner: Chuck O. Kendall
	)	
Appl. No.: 09/783,250	)	
	)	
Applicant's Docket:	)	
JP920000411US1	)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir:

REAL PARTY IN INTEREST

The assignee, International Business Machines Corporation, is the real party in interest.

RELATED APPEALS AND INTERFERENCES

This is the first appeal in the present patent application. There are no other appeals or interferences known to the appellant or its legal representative. International Business Machines Corporation is the sole assignee of the patent application.

STATUS OF CLAIMS

Claims 1-45 were originally presented in the application. In a first Office action dated September 25, 2003, (the "First Office Action") all the claims were rejected. Specifically, claims 1-6, 9-15, 17-23, 25-36 and 38-44 were rejected under 35 U.S.C. 102(e) based on U.S. patent 6,067,639 (Rodrigues). Claims 7, 8, 24, 37 and 45 were rejected under 35 U.S.C. 103(a) based on Rodrigues in view of U.S. patent 5,860,009 (Uchihira). Claim 16 was rejected under 35 U.S.C. 103 (a) based on Rodrigues et al. in view of U.S. patent 6,397,378 B1 (Grey).

In response to the First Office Action dated December 26, 2003, method claims 1, 2, 3 and 5, computer program product claims 18, 19, 20 and 22, and system claims 31, 32, 33 and 35 were responsively amended in an effort to overcome the rejections. Claims 23 and 36 were also amended to conform them to their amended base claims. Claims 9, 25 and 38 were amended to correct informalities.

In an Office action of February 10, 2004 (the "Final Office Action"), claims 1-45 were finally rejected. Specifically, claims 1-6, 9-16, 17-23, 25-36 and 38-44 were rejected under 35 U.S.C. 103(a) based on Rodrigues in view of U.S. patent 6,067,639 (Darty). Claims 7, 8, 24, 37 and 45 were rejected under 35 U.S.C. 103(a) based on Rodrigues in view of Darty, and further in view of Uchihira.

In a Notice of Appeal received by the USPTO on June 14, 2004, Appellant appealed from the final rejection of the claims.

STATUS OF AMENDMENTS

No claims have been allowed.

In reviewing the reply of December 26, 2003, for preparation of this Appeal Brief, Appellant has discovered that claim 28 was submitted with an incorrect reference to itself. (This incorrect reference was not objected to in the First Office Action or the Final Office Action.) Amendment is herein submitted to claim 28 to correct its dependency, so that claim 28 depends on claim 29 instead of itself.

Also, Appellant has determined that claim 26 lacks antecedent basis for "an argument comprising an instance of another object," as in claims 11 and 40. (This was not objected to in

the First Office Action or the Final Office Action.) Claim 26 is herein amended to provide this antecedent basis.

Also, Appellant has determined that for claims 10 and 39 "said argument" lacks agreement in number with regard to "arguments" in claims 9 and 38, respectively. (This was not objected to in the First Office Action or the Final Office Action.) Claims 10 and 39 are herein amended to refer to "such an argument," which is in agreement with "arguments" of claims 9 and 38.

The claims set out in Appendix "AA" herein below reflect the amendments as entered responsive to Appellant's reply of December 26, 2003, and as submitted herein.

SUMMARY OF INVENTION

The present invention is claimed in the form of a method, a computer program product and a system in independent claims 1, 18 and 31, respectively. Claim 1, for example, points out that a method of testing a program includes step a), according to which the program is divided into groups *such that every statement in the program belongs to at least one of the groups*. (The discussion herein of claim 1 also applies to claims 18 and 31. That is, independent claims 1, 18 and 31 have similar language, each according to the forms of the invention they claim.) This limitation was added to the claims in reply to the First Office Action because it is an aspect of how there is a determination based on executed groups that *all* statements have been executed, and because this aspect is not taught by the cited art. Support for this aspect of the invention is found in the specification and figures. See page 18 and FIG's 4A - 4C (showing all statements in the program of FIG. 4A, i.e., lines 3, 5, 6, 8 - 18, 20 - 25, 27 and 28, divided into respective groups B11 - B18 in FIG's 4B and 4C, wherein group B11 in FIG. 4B includes lines corresponding to lines 3, 5 and 6 of FIG. 4A, group B12 includes lines corresponding to lines 8 and 9 of FIG. 4A, group B13 in FIG. 4B includes a line corresponding to line 10 of FIG. 4A, group B14 in FIG's 4B and 4C includes lines corresponding to lines 11 - 13 of FIG. 4A, group B15 in FIG. 4C includes lines corresponding to lines 14 - 15 of FIG. 4A, group B16 in FIG. 4C includes lines corresponding to lines 16 - 18 of FIG. 4A, group B17 in FIG. 4C includes lines corresponding to lines 20 - 21 of FIG. 4A, and group B18 in FIG. 4C includes lines corresponding to lines 22 - 25, 27 and 28 of FIG. 4A).

Claim 1 goes on to state that *each of the groups contains a respective sequence of ones of the statements* such that all the statements of *such a group* are executed if at least one statement of said group is executed. The claim was amended to particularly point out this feature of the invention in order to make it clear that it is not the plurality of groups that has the sequence of statements. Rather, each individual group has its own sequence of statements. (It should be understood that the "sequence of . . . statements" is not necessarily the same for each group. That is, each group may have its own respective sequence of statements.) Support for this feature is found in the specification and figures. See above reference to page 18 and FIG's 4A - 4C (showing sequences such as the statement of line 10 of FIG. 4A for group B13 (a sequence of one statement) and the statements of lines 11 - 13 of FIG. 4A for group B14, etc.). Claim 1 goes on to state that such a group is deemed to be executed if at least one of the statements of the group is executed when the program is executed. For support, refer again to page 18 and FIG's 4A - 4C (showing groups of statements in which none of the groups have any branching statements, except at the end of the group, that would cause some of the statements in the group to not be executed along with the others in the group). With this change, a clearer meaning is provided regarding the executed group referred to in the next step.

Claim 1 goes on to set out step b), according to which there is a determining of *the ones of the groups* that are executed when said program is executed while testing said program. See "Overview and Discussion of the Invention," beginning on page 8, "Method to Ensure Coverage in Testing," beginning on page 13, and FIG. 2. With this language, which was added in reply to the First Office Action, step b) is more explicitly tied back to the earlier step a) in such a way that now step b) is clearly not anticipated by art that merely discloses a determination about just any kind of "groups" that are executed. That is, a "group" as now mentioned in step b) is more clearly the particular kind of group defined in the preceding step a).

Claim 1 goes on to set out step c). In this step unexecuted ones of the groups are indicated based on the ones of the groups that were determined *in step b)* to have been executed. This more clearly points out the particular invention than did the language of the claim as originally submitted, which stated that "unexecuted groups are determined based on said determining." That is, claim 1 makes it clear that the unexecuted groups are determined based on the earlier step, which determined the groups that have been executed.

Finally, claim 1 states that in step d) a tester is enabled to execute said unexecuted groups such that said tester can ensure that all statements in said program are executed at least once.

Claims 2, 19 and 32 indicate an extra statement is included in each of said groups, wherein execution of such an extra statement enables said determining in step b) to identify an executed one of the groups corresponding to said extra statement. See page 15, lines 9-10.

Claims 3, 20 and 33 state that said extra statements contain respective group identifiers, wherein said determining in step b) further comprises examining such a group identifier to determine a specific one of the groups which has been executed. See page 15, lines 10-14.

Claims 5, 22 and 35 state that the method includes grouping a sequence of the groups into a block; and determining that said block has been executed only if all of the groups of the block are executed. See page 19, lines 10-15.

ISSUES

Are claims 1-6, 9-16, 17-23, 25-36 and 38-44 unpatentable under 35 U.S.C. 103(a) in view of the combination of Rodrigues and Darty?

Are claims 7, 8, 24, 37 and 45 unpatentable under 35 U.S.C. 103(a) in view of the combination of Rodrigues in view of Darty, and further in view of Uchihira?

GROUPING OF CLAIMS

Solely for the purpose of this appeal, the claims stand or fall together according to the following groups:

Group 1: claims 1, 18 and 31, and also claim 45;

Group 2: claims 2, 19 and 32; and also claims 4, 9, 13-17, 21, 25, 29, 34, 38 and 42-44;

Group 3: claims 3, 20 and 33;

Group 4: claims 5, 22 and 35; and also claims 6-8, 23, 24, 36 and 37; and

Group 5: claims 10-12, 26-28, 30 and 39-41.

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001

ARGUMENT

<u>Claims 1, 18 and 31, and also claim 45.</u>

The Final Office Action rejects claims 1, 18 and 31 on the basis that they are obvious in view of Rodrigues and Darty. Appellant respectfully disagrees. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)). Applicant contends that claims 1, 18 and 31, as amended in reply to the First Office Action are patentably distinct because all the claim limitations are not taught or suggested by the prior art.

The Final Office Action cites Darty, a new reference, asserting that Darty teaches what Rodrigues fails to teach with respect to features claimed for the invention of the present patent application. Aside from whether Darty truly does teach the claimed feature the Final Office Action contends that it does, the Final Office Action continues to assert that Rodrigues teaches certain features claimed for the invention of the present patent application and relies upon these asserted features for the rejection of claims 1, 18 and 31 in the present case, but does not substantially respond to numerous arguments presented in reply to the First Office Action. Instead, the Final Office Action merely recites the amended claims in the present application and points to the same passages in Rodrigues that were relied upon in the First Office Action for the rejection of the original claims.

Rodrigues largely concerns randomly selecting test operations and logging the randomly generated sequence of operations in a playback file. See, for example, column 4, lines 52 - 67; column 9, lines 18 - 49. In this regard, Rodrigues teaches about weighting test object groups to influence how they are *randomly* selected. Rodrigues, column 16, lines 1 -15. Rodrigues does also teach about manual playback creation, but never suggests dividing a program into a plurality of groups such that every statement in the program belongs to at least one of the groups.

The Final Office Action cites column 15, lines 54 - 65, of Rodrigues regarding the claimed feature in step a) in the present case, "dividing said program into a plurality of groups such that every statement in the program belongs to at least one of the groups." However, the cited portion of Rodrigues concerns weighting test object groups to influence how they are *randomly* selected, as is elaborated upon in the immediately following passage of Rodrigues,

i.e., column 16, lines 1 -15. This does not teach or suggest dividing a program being tested into groups of statements such that every statement in the program belongs to at least one of the groups, as claimed.

The Final Office Action cites Rodrigues, FIG. 7, elements 704 and 706 regarding the claimed feature of step b) in the present case, "determining the ones of the groups that are executed when said program is executed while testing said program," and cites column 13, lines 28 - 35, of Rodrigues regarding the claimed feature of step c) in the present case, "indicating unexecuted ones of the groups based on the ones of the groups that were determined in step b) to have been executed." However, the cited portions of Rodrigues do not concern *groups*, but rather concern keeping track of "entries" or "test operation objects" that were randomly selected, to ensure they all are played back.

Moreover, claim 1 of the present application states that "each of said groups contains a respective sequence of ones of the statements such that all the statements of such a group are executed if at least one statement of said group is executed, and wherein such a group is deemed to be executed if at least one of the statements of the group is executed when the program is executed." The Final Office Action contends that Darty FIG. 3A, elements S102-S108, and FIG. 3B, element S122, provides this teaching and that there is motivation to combine Rodrigues with Darty in this regard.

Darty teaches, in relevant part regarding the description of the cited FIG's and elements, that "lines of code of the program to be tested are grouped into functional blocks." Darty, col. 9, lines 39-41. Darty states that test points are associated with code blocks. Darty, col. 2, lines 25-28; see also col. 9, lines 28-32. Each such test point includes an output statement in the code "to determine program execution status." Darty, col. 9, lines 28-30. Flow chart FIG. 3B, element S122, merely states that each test point is associated with a respective block of code. None of this teaches or suggests that that all the statements of a test point's associated code block are executed if at least one statement of said group is executed.

Even further, the Final Office Action must provide a legitimate basis for the suggestion or motivation to combine the references, as required for a rejection of this kind. MPEP 2143.01 ("Suggestion or Motivation To Modify the References"). However, the basis asserted in the Final Office Action is erroneous. The Final Office Action asserts it would have been obvious to

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001

combine the references "because using a particular sequence of test instruction[s] for a particular group would make testing the divisions more efficient." Final Office Action, page 3, first paragraph. In the first place, this conclusory statement addresses only a slight aspect of the claimed invention, and does not sufficiently justify the combination of the references.

Even more importantly, Darty actually *teaches away* from the teaching for which Rodrigues is relied upon, i.e., that *all* lines of code in the program are grouped into functional blocks.[1] Thus neither Rodrigues nor Darty suggests the desirability of the combination. MPEP 2143.01 (citing *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990) regarding the point that just because references *can* be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination).

Darty states that test points are associated with code blocks. Darty, col. 2, lines 25-28; see also col. 9, lines 28-32. The test points are "strategically placed" so that a block of code that is the "most probable" source of a failure may be identified, "depending on the number and placement of test points." Darty, col. 9, lines 24-38; see also col. 10, lines 3-10. This is contrary to the claims in the present application, that every line of code is assigned to a block and execution of every block is ensured. If Darty taught that every line was assigned to a block and every block was executed, then Darty would not discuss probabilities about which one of the blocks gives rise to an execution failure. Thus, the combination of Darty with Rodrigues is not indicated for the teachings asserted in the Final Office Action.

From the above analysis it should be appreciated that neither Rodrigues nor Darty address the issue of ensuring that *all statements in an entire program* are executed at least once, as per claims 1, 18 and 31 of the present application. Accordingly, neither Rodrigues nor Darty, alone or in combination, teach or suggest creating a set of groups that constitute an exhaustive collection of all statements in a program under test, keeping track of the executed ones of the groups and then forcing execution of the statements any such group that has not executed, as per the claims of the present application.

Claim 45 depends on claim 31, which is patentably distinct, as discussed herein above. This basis alone is sufficient for patent ability. MPEP 2143.03 ("If an independent claim is non

---

[1] Setting aside the point made herein above, that Rodrigues does not actually supply the teaching for which it is relied upon.

obvious under 35 U.S.C. 103, then any claim depending therefrom is non obvious," citing *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)).

Claims 2, 19 and 32; and also claims 4, 9, 13-17, 21, 25, 29, 34, 38 and 42-44.

The Final Office Action rejects claims 2, 19 and 32 on the basis that they are obvious in view of Rodrigues and Darty. Appellant respectfully disagrees. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)). Applicant contends that claims 2, 19 and 32, as amended in reply to the First Office Action are patentably distinct because all the claim limitations are not taught or suggested by the prior art.

Despite amendments and arguments presented in reply to the First Office Action, the Final Office Action merely recites the amended claims in the present application and points to the same passages in Rodrigues that were relied upon in the First Office Action for the rejection of the original claims.

Claim 2 points out that the method also particularly includes an extra statement in each of the previously mentioned groups, and that the execution of such an extra statement enables the determining in step b) to identify an executed one of the groups corresponding to the extra statement. (The discussion herein of claim 2 also applies to claims 19 and 32.) That is, identifying that the extra statement is executed enables identifying an executed *group* (and, consequently, all the group's associated statements). See page 18 and FIG's 4A - 4C ("Instrumentation module 310 may insert program statements . . . While only one program statement is shown inserted for each group, more than one statement also can be inserted as a designer wishes. The effect of the inserted program statement is to pass the group identifier in the statement to coverage tracking module 350 when the statement is executed.").

In rejecting these claims, the Final Office Action merely points again at column 15, lines 60 - 62, of Rodrigues. As stated above in connection with claim 1, the cited portion of Rodrigues concerns defining groups of *test objects* and weighting the test object groups to influence how they are *randomly* selected. This does not teach or suggest dividing a program being tested into groups of statements such that every statement in the program belongs to at least one of the

groups, as stated in claim 1, and it does not teach or suggest including an extra statement in each of such group of program statements, as stated in claim 2 of the present application.

Claims 4, 9, 13-17, 21, 25, 29, 34, 38 and 42-44 all depend on claims that are patentably distinct, as discussed herein above. This basis alone is sufficient for patent ability. MPEP 2143.03 ("If an independent claim is non obvious under 35 U.S.C. 103, then any claim depending therefrom is non obvious," citing *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)).

Claims 3, 20 and 33.

The Final Office Action rejects claims 3, 20 and 33 on the basis that they are obvious in view of Rodrigues and Darty. Appellant respectfully disagrees. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)). Applicant contends that claims 3, 20 and 33, as amended in reply to the First Office Action are patentably distinct because all the claim limitations are not taught or suggested by the prior art.

Despite amendments and arguments presented in reply to the First Office Action, the Final Office Action merely recites the amended claims in the present application and points to the same passages in Rodrigues that were relied upon in the First Office Action for the rejection of the original claims.

Claim 3 states that the extra statements contain respective group identifiers, and that the determining in step b) includes examining such a group identifier to determine a specific one of the groups which has been executed. (The discussion herein of claim 3 also applies to claims 20 and 33.) The language of claim 3 makes it clear that each group has its own respective group identifier, and that the "groups" referred to are the *same particular kind of groups* defined in the earlier claims. See page 18 and FIG's 4A - 4C ("Instrumentation module 310 may insert program statements, with each statement containing class name and group identifier as parameters into program code as depicted, for example, in lines 3, 9, 14, 18 of Figure 4B.").

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001

In rejecting these claims, the Final Office Action yet again relies upon Rodrigues column 15, lines 60 - 62, and also relies upon column 16, lines 1 - 15. Both of these passages are discussed herein above in connection with claims 1 and 2.

Claims 5, 22 and 35; and also claims 6-8, 23, 24, 36 and 37.

The Final Office Action rejects claims 5,22 and 35 on the basis that they are obvious in view of Rodrigues and Darty. Appellant respectfully disagrees. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)). Applicant contends that claims 5, 22 and 35, as amended in reply to the First Office Action are patentably distinct because all the claim limitations are not taught or suggested by the prior art.

Despite amendments and arguments presented in reply to the First Office Action, the Final Office Action merely recites the amended claims in the present application and points to the same passages in Rodrigues that were relied upon in the First Office Action for the rejection of the original claims.

Claim 5 states that the method also involves grouping a *sequence of the groups* into a block. (The discussion herein of claim 5 also applies to claims 22 and 35.) Also, it states that the method includes determining that the block has been executed only if all of the groups of the block are executed. Once again, the "groups" referred to are not just any kind of groups, but rather are the *same particular kind of groups* defined in the earlier claims.

In rejecting these claims the Final Office Action relies upon Rodrigues column 9, lines 62-65. This portion of Rodrigues concerns the *test object* groups referred to in the previously cited portions. See the Applicant's response above.

Claims 6-8, 23, 24, 36 and 37 all depend on claims that are patentably distinct, as discussed herein above. This basis alone is sufficient for patent ability. MPEP 2143.03 ("If an independent claim is non obvious under 35 U.S.C. 103, then any claim depending therefrom is non obvious," citing *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)).

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001
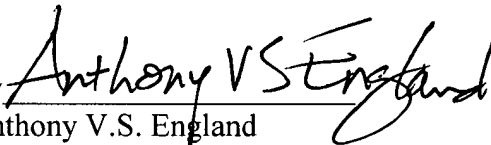
Group 6: claims 10-12, 26-28, 30 and 39-41.

Claims 10-12, 26-28, 30 and 39-41 all depend on claims that are patentably distinct, as discussed herein above. This basis alone is sufficient for patent ability. MPEP 2143.03 ("If an independent claim is non obvious under 35 U.S.C. 103, then any claim depending therefrom is non obvious," citing *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)).

These claims are grouped separately because these are claims for which amendments are herein submitted, or else they depend upon claims for which amendments are herein submitted. Therefore, their allowability might depend upon granting the entry of the amendments submitted herein.

REQUEST FOR ACTION

Based on the above arguments, Appellant requests that claims 1-45 of the present application be allowed and the application promptly be passed to issuance.

Respectfully submitted,

By

Anthony V.S. England
Registration No. 35,129
Attorney of Record for
IBM Corporation
1717 West Sixth Street, Suite 230
Austin, Texas 78703
Telephone: 512-477-7165
a@aengland.com

ATTACHMENTS: APPENDIX "AA" CLAIMS

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001

APPENDIX "AA"

What is claimed is:

1. (previously presented) A method of testing a program having statements, said method comprising the steps of:

a) dividing said program into a plurality of groups such that every statement in the program belongs to at least one of the groups, wherein each of said groups contains a respective sequence of ones of the statements such that all the statements of such a group are executed if at least one statement of said group is executed, and wherein such a group is deemed to be executed if at least one of the statements of the group is executed when the program is executed;

b) determining the ones of the groups that are executed when said program is executed while testing said program;

c) indicating unexecuted ones of the groups based on the ones of the groups that were determined in step b) to have been executed; and

d) enabling a tester to execute said unexecuted groups such that said tester can ensure that all statements in said program are executed at least once.

2. (previously presented) The method of claim 1, further comprising including an extra statement in each of said groups, wherein execution of such an extra statement enables said determining in step b) to identify an executed one of the groups corresponding to said extra statement .

3. (previously presented) The method of claim 2, wherein said extra statements contain respective group identifiers, wherein said determining in step b) further comprises examining such a group identifier to determine a specific one of the groups which has been executed.

4. (original) The method of claim 2, wherein said program is contained in a plurality of programs which in turn are contained in a class of an object oriented environment.

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001

5. (previously presented) The method of claim 4, further comprising the steps of:

grouping a sequence of the groups into a block; and

determining that said block has been executed only if all of the groups of the block are executed.

6. (original) The method of claim 5, wherein said grouping comprises:

determining a language structure present in said plurality of programs;

grouping a subset of groups present in said language structure into a block such that the statements in said language structure are presented as a block to said tester.

7. (original) The method of claim 6, wherein said blocks are defined hierarchically according to the inclusive relationship of language structures in said plurality of programs.

8. (original) The method of claim 7, wherein said language structure comprises one of program delimiters, control structure and loop structure.

9. (previously presented) The method of claim 4, wherein said enabling comprises:

enabling said tester to examine the statements associated with said unexecuted blocks such that said tester can determine arguments which would cause an unexecuted block to be executed;

enabling said tester to enter said determined arguments to cause said unexecuted block to be executed.

10. (currently amended) The method of claim 9, wherein such an ~~said~~ argument comprises an instance of another object.

11. (original) The method of claim 10, further comprises:

enabling said tester to instantiate said instance of said another object;

enabling said tester to assign a name to said instance, wherein said tester can enter said name to provide said instance as an argument value.

12. (original) The method of claim 11, further comprising:

receiving a string as an argument; and

determining that said string indicates that said instance is said argument value if said

name matches said string.

13. (original) The method of claim 4, further comprising:

enabling said tester to define a macro containing a plurality of program lines;

storing said macro in a database; and

enabling said tester to execute said macro in the middle of testing said plurality of

programs.

14. (original) The method of claim 13, wherein said macro is designed to examine the

data structures within an instance of an object or to set the values for the variables in the object.

15. (original) The method of claim 4, wherein said dividing, determining, indicating and

enabling are performed in a single computer system.

16. (original) The method of claim 4, wherein said object is generated in Java

Programming language.

17. (original) The method of claim 4, further comprising:

enabling said tester to load said class;

enabling said tester to instantiate an instance of said class; and

enabling said tester to execute said program on said instance.

18. (previously presented) A computer program product for use with a computer system, said computer program product comprising a computer usable medium having computer readable program code means embodied in said computer usable medium, said computer readable program code means enabling testing of a program having statements, said computer readable program code means comprising:

dividing means for dividing said program into a plurality of groups such that every statement in the program belongs to at least one of the groups, wherein each of said groups contains a respective sequence of ones of the statements such that all the statements of such a group are executed if at least one statement of said group is executed, and wherein such a group is deemed to be executed if at least one of the statements of the group is executed when the program is executed;

determining means for determining the ones of the groups that are executed when said program is executed while testing said program;

indicating means for indicating unexecuted ones of the groups based on said determining; and

enabling means for enabling a tester to execute said unexecuted groups such that said tester can ensure that all statements in said program are executed at least once.

19. (previously presented) The computer program product of claim 18, further comprising including means for including an extra statement in each of said groups, wherein execution of such an extra statement enables said determining means to identify an executed one of the groups corresponding to said extra statement.

20. (previously presented) The computer program product of claim 19, wherein said extra statements contain respective group identifiers, wherein said determining means examines such a group identifier to determine a specific one of the groups which has been executed.

21. (original) The computer program product of claim 19, wherein said program is contained in a plurality of programs which in turn are contained in a class of an object oriented environment.

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001

22. (previously presented) The computer program product of claim 21, further comprising grouping means for grouping a sequence of the groups into a block, and second determining means for determining that said block has been executed only if all of the groups of the block are executed.

23. (previously presented) The computer program product of claim 22, wherein said grouping means comprises:

third determining means for determining a language structure present in said plurality of programs;

a second grouping means for grouping a subset of groups present in said language structure into a block such that the statements in said language structure are presented as a block to said tester.

24. (original) The computer program product of claim 23, wherein said blocks are defined hierarchically according to the inclusive relationship of language structures in said plurality of programs.

25. (previously presented) The computer program product of claim 21, wherein said enabling means comprises:

second enabling means for enabling said tester to examine the statements associated with said unexecuted blocks such that said tester can determine arguments which would cause an unexecuted block to be executed;

third enabling means for enabling said tester to enter said determined arguments to cause said unexecuted block to be executed.

26. (currently amended) The computer program product of claim 25~~21~~, wherein such an argument comprises an instance of another object, and the computer program product further comprises:

means for enabling said tester to instantiate said instance of said another object;

means for enabling said tester to assign a name to said instance, wherein said tester can enter said name to provide said instance as an argument value.

27. (original) The computer program product of claim 26, further comprising:

means for receiving a string as an argument; and

means for determining that said string indicates that said instance is said argument if said name matches said string.

28. (currently amended) The computer program product of claim 29~~28~~, wherein said macro is designed to examine the data structures within an instance of an object or to set the values for the variables in the object.

29. (original) The computer program product of claim 21, further comprising:

second enabling means for enabling said tester to define a macro containing a plurality of program lines;

storing means for storing said macro; and

third enabling means for enabling said tester to execute said macro in the middle of testing said plurality of programs.

30. (original) The computer program product of claim 26, further comprising:

means for enabling said tester to load said class;

means for enabling said tester to instantiate an instance of said class; and

means for enabling said tester to execute said program on said instance.

31. (previously presented) A system enabling a tester to test a program having statements, said computer system comprising:

a random access memory (RAM);

a display unit containing a display screen;

an input interface;

a processor dividing said program into a plurality of groups such that every statement in the program belongs to at least one of the groups, wherein each of said groups contains a respective sequence of ones of the statements such that all the statements of such a group are executed if at least one statement of said group is executed, and wherein such a group is deemed to be executed if at least one of the statements of the group is executed when the program is executed,

said processor executing said program in response to instructions received from said input interface,

said processor determining the ones of the groups that are executed when said program is executed,

said processor causing a display to be generated on said display unit, said display indicating unexecuted ones of the groups based on the ones of the groups that were determined to have been executed; and

said processor enabling said tester to execute said unexecuted groups such that said tester can ensure that all statements in said program are executed at least once.

32. (previously presented) The system of claim 31, wherein said processor includes an extra statement in each of said groups, wherein execution of such an extra statement enables said processor to identify an executed one of the groups corresponding to said extra, said computer system further comprising a secondary storage wherein said processor stores said program including said extra statement on said secondary storage.

33. (previously presented) The system of claim 32, wherein said extra statements contain respective group identifiers, wherein said processor examines such a group identifier to determine a specific one of the groups which has been executed.

34. (original) The system of claim 32, wherein said program is contained in a plurality of programs which in turn are contained in a class of an object oriented environment.

35. (previously presented) The system of claim 34, wherein said processor groups a sequence of the groups into a block, and wherein said display indicates that said block has been executed only if all of the groups of the block are executed.

36. (original) The system of claim 35, wherein said processor groups said sequence of groups according to a language structure present in said plurality of programs.

37. (original) The system of claim 36, wherein said blocks are defined hierarchically according to the inclusive relationship of language structures in said plurality of programs.

38. (previously presented) The system of claim 34, wherein said processor receives instructions from said input interface to display the statements associated with said unexecuted blocks, said processor causing the statements to be displayed on said display unit such that said tester can determine arguments which would cause an unexecuted block to be execute.

39. (currently amended) The system of claim 38, wherein such ansaid argument comprises an instance of another object.

40. (original) The system of claim 39, wherein said processor instantiates said instance of another object in response to receiving an instruction to instantiate said instance of said another object, said processor further associating a name associated with said instance of another object, wherein said name is received from said input interface and said tester can enter said name to provide said instance as an argument value.

Docket JP920000411US1

Appl. No.: 09/783,250
Filed: February 14, 2001

41. (original) The system of claim 40, wherein said processor receives a string as an argument and determines that said string indicates that said instance is said argument value if said name matches said string.

42. (original) The system of claim 34, wherein said processor receives a plurality of program lines representing a macro, said processor storing said macro in a database, said processor executing said macro in response to receiving an instruction to execute said macro.

43. (original) The system of claim 42, wherein said macro is designed to examine the data structures within an instance of an object or to set the values for the variables in the object.

44. (original) The system of claim 34, wherein said processor loads said class into said RAM in response to receiving an instruction to load said class, said processor further instantiating an instance of said class in response to receiving another instruction, said processor executing said program on said instance in response to receiving one more instruction.

45. (original) The system of claim 31, wherein said input interface is connected to at least one of a mouse and a key-board.
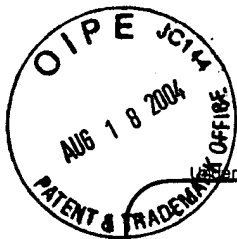
TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/783,250
Filing Date	02/14/2001
First Named Inventor	Kallol Pal
Art Unit	2122
Examiner Name	Chuck O. Kendall
Attorney Docket Number	JP920000411US1

Total Number of Pages in This Submission **65**

ENCLOSURES   *(Check all that apply)*

- [✔] Fee Transmittal Form
  - [✔] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
  - [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53

- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____

- [ ] After Allowance communication to Technology Center (TC)
- [ ] Appeal Communication to Board of Appeals and Interferences
- [✔] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief*)**
- [ ] Proprietary Information
- [ ] Status Letter
- [✔] Other Enclosure(s) (please Identify below):

Acknowledgement postcard

Remarks
*in triplicate (Appeal Brief **12** pages, Appendix "AA" **9** pages))

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Anthony V.S. England
Signature	*Anthony VS England*
Date	8/14/2004

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name	Anthony V.S. England	
Signature	*Anthony VS England*	Date 8/14/2004

FEE TRANSMITTAL
for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($) 330.00

Complete if Known

Application Number	09/783,250
Filing Date	02/14/2001
First Named Inventor	Kallol Pal
Examiner Name	Chuck O. Kendall
Art Unit	2122
Attorney Docket No.	JP920000411US1

METHOD OF PAYMENT (check all that apply)

☐ Check   ☐ Credit card   ☐ Money Order   ☐ Other   ☐ None

☑ Deposit Account:

Deposit Account Number: **09-0457**

Deposit Account Name: **International Business Ma**

The Director is authorized to: (check all that apply)

☑ Charge fee(s) indicated below   ☑ Credit any overpayments

☑ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Fee ($)	Small Entity Fee Code	Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

	Extra Claims	Fee from below	Fee Paid
Total Claims	-20** =	X	=
Independent Claims	- 3** =	X	=
Multiple Dependent			=

Large Entity Fee Code	Fee ($)	Small Entity Fee Code	Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0

**or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Fee ($)	Small Entity Fee Code	Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	$330.00
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 330.00

SUBMITTED BY			(Complete *(if applicable)*)
Name (Print/Type)	Anthony V.S. England	Registration No. (Attorney/Agent) 35,129	Telephone 512-477-7165
Signature	*Anthony VS England*		Date 8/14/2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.